

REMARKS/ARGUMENTS

This Amendment is in response to the Office Action mailed July 26, 2006. Claims 4-88 were pending in the present application. This Amendment amends claims 4, 8, 20, 27, 33, 38, 43, 44, 47, 51, 56, 62, 66, 72, 81, 86, 87, and 88, without adding or canceling any claims, leaving pending in the application claims 4-88. Reconsideration of the rejected claims is respectfully requested.

I. Double Patenting Rejection

Claims 4, 6, 8, 17-21, 31-33, 35-36, 38, 40, 43, 53, 56, 58, 60-63, 66, 77, 79-81, 84, and 86-88 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being obvious over claims 4, 7, 5, 11, 12, 13, 14, 15, 24, 25, 27, 29, 31-33, 42, 48, 51, 52-54, 56, 65, 67, 68, and 70-73, of U.S. Patent Application No. 10/035,413. It is respectfully submitted that the cited claims of U.S. Patent Application No. 10/035,413 recite a "user customizable immediate access keystroke combination," which allows a user to customize the functionality of an application user interface. The cited claims of the present application recite a "customizable view all command," which allows a user to select a view to be displayed to the user each time the user accesses a user interface. It is respectfully submitted that allowing a user to customize keystroke combinations would not render it obvious to also customize a view all command. It is respectfully submitted that these concepts are patentably distinct, and that allowing the user to change the interaction of the keyboard with the user interface, such as by adding a customizable immediate access keystroke combination, does not make it obvious to utilize a customizable view all command to persistently set the view of a user interface for a specific user. As such, Applicants respectfully request that the provisional rejection with respect to these claims be withdrawn.

II. Rejection under 35 U.S.C. §103

Claims 4-88 are rejected under 35 U.S.C. §103(a) as being obvious over *Anuff* (US 6,327,628) in view of *Hargrove* (US 5,371,847) and *D'Arlach* (US 6,026,433). Applicants respectfully submit that these references do not teach or suggest each element of these claims.

For example, Applicants' claim 4 as amended recites a system for generating a customizable application user interface, including:

means for allowing a user to specify configuration data to customize a customizable view all command in the application user interface, the configuration data defining a property of a user interface element for the view all command, the customizable view all command further being associated with metadata characterizing the user interface element;

a data repository including a data record and configured to store the configuration data in the data record, the configuration data defining a current state of the property;

an application user interface generator configured to generate the application user interface in response to a request from a client device, the user interface generator being operable to access the data record to determine the current state of the customizable view all command at substantially the time of the request and generate markup language including the current state of the user customizable view all command;

a web application server operable to deliver the generated markup language for the application user interface to the client device; and

means for preserving the current state of the customizable view all command between one generation of the application user interface and another generation of the application user interface, the application user interface generator being further configured to automatically generate the one generation of the application user interface and the another generation of the application user interface as configured with the property of the view all command for the user interface element

(*emphasis added*). Such limitations are neither taught nor suggested by these references.

For example, *Anuff* teaches the customization of various aspects of the layout of a plurality of modules in a portal page (col. 2, lines 1-12). The user is able to customize such aspects as the layout, color scheme, selection, display content, and order of modules that are displayed (col. 3, line 40-col. 4, line 14; col. 14, lines 3-14). *Anuff* thus teaches allowing a user to modify display aspects of a portal page, but does not teach or suggest allowing a user to control functional aspects of a user interface, such as functional aspects of a user interface element. Further, the Office Action recognizes on page 4 that *Anuff* fails to teach receiving input to define a property of a customizable element and then automatically generate a user interface containing that customizable element. *Anuff* also does not teach or suggest accessing a data record in response to a request from a user to determine a current state of a user interface element as selected by a user, then generating markup language in response thereto containing the current state. *Anuff* further fails to teach or suggest a customizable view all command handled in this way, as is further recognized on page 4 of the Office Action. For at least these reasons, *Anuff* cannot render obvious Applicants' claim 4 or the claims that depend therefrom. The other independent claims recite limitations that similarly are not taught or suggested by *Anuff*, such

that these claims and the claims that depend therefrom similarly are not rendered obvious by *Anuff*.

Hargrove does not make up for the deficiencies in *Anuff* with respect to these claims. *Hargrove* teaches a system for specifying the arrangement of windows on a display (col. 2, lines 18-52), and is cited as teaching a view all command (OA p. 4). *Hargrove* does not, however, teach a customizable view all command, or how to customize a view all command. Further, *Hargrove* does not teach or suggest allowing a user to control functional aspects of a user interface, such as functional aspects of a user interface element. *Hargrove* fails to teach receiving input to define a property of a customizable element and then automatically generate a user interface containing that customizable element. *Hargrove* also does not teach or suggest accessing a data record in response to a request from a user to determine a current state of a user interface element as selected by a user, then generating markup language in response thereto containing the current state. As such, *Hargrove* cannot render obvious these claims, either alone or in combination with *Anuff*.

Even if, for purposes of argument, *Hargrove* teaches a view all command, and it were obvious to combine this command with *Anuff*, the combination would simply allow the interface of *Anuff* to control the display location or inclusion of the view all command, for example, and would not allow a user to change a functional property of the command. Further, the combination would not check a data record in response to a request from a user to determine the current state of the view all command, then generate markup language in response thereto for delivery to the user. As such, a combination of *Hargrove* and *Anuff* still would not render obvious these claims.

D'Arlach does not make up for these deficiencies in *Hargrove* and *Anuff* with respect to these claims. *D'Arlach* is cited as teaching generation of an application user interface using a property of a user interface element (OA p. 4). *D'Arlach*, a reference dating back to 1997, discloses the use of a plurality of customizable templates that allow a user to easily create and maintain a Web site "without requiring knowledge of HTML," allowing the user to author a Web site from a client computer (col. 2, lines 30-49). A customizable template contains customizable objects or elements such as buttons and text labels, which in turn have attributes or properties associated therewith (col. 4, lines 59-67). A user can create a Web site by selecting various

elements and attributes associated with a selected template (col. 5, lines 54-65). After creating or editing the site, the site is published (col. 6, lines 44-46). "Publishing" the site means that a directory is generated in the server and the associated files (including HTML files) are "generated" and stored in the directory, which can be used to serve subsequent user requests (col. 6, lines 54-60). The creation of Web pages using templates and the generation of HTML through publishing is well known in the art. For each user to have a customized site, each user must pull up the templates, modify the elements and attributes, and then publish the site to a directory. Each user will have a unique set of HTML documents generated at the time of publishing, and anyone accessing that user's site in the future will receive those HTML documents.

This is very different than what is recited in Applicants' pending claims. Applicants' disclosure is directed to a user-customizable interface to a Web based application, for example. A user can personalize options and functionality to be presented in the interface, and data for the personalization is stored in a database. A unique set of HTML documents is not generated for each customized user, as in *D'Arlach*. When a user submits a request, a user interface generator, for example, will access the saved configuration data to determine a current state of a user interface element and generate markup-language data, such as HTML data, suitable for presentation as a Web page (paragraph [0046]). Therefore, a unique set of HTML documents is generated in response to the request from the user, such that each user can receive a customized interface without having to generate and store separate pages for each user as in *D'Arlach*.

As discussed above, Applicants' claims recites a data repository or data record including information associated with the customizable property, the information corresponding to a current state of the user interface element being modifiable by a user. *D'Arlach* does not disclose such a limitation, as *D'Arlach* instead requires a new HTML page to be generated ("published") in order to modify an element or attribute of the page. *D'Arlach* does not allow for the updating of a data record storing an attribute to be used for generating the page.

Applicants' claims also recite a user interface generator configured to generate the application user interface in response to a request from a user, and to use the data record to determine a current state of the user customizable property at substantially the time of the request and generate markup language including the current state of the user customizable property. *D'Arlach* does not disclose or suggest such a limitation, as *D'Arlach* instead generates specific

HTML files for each user before any requests are received or served. Further, *D'Arlach* does not teach or suggest a user customizable view all property, or means for customizing such a user interface element.

As *D'Arlach* does not teach or suggest such limitations, *D'Arlach* cannot render obvious these claims, either alone or in any combination with *Hargrove* and *Anuff*.

Applicants therefore respectfully request that the rejections with respect to claims 4-88 be withdrawn.

III. Amendment to the Claims

Unless otherwise specified, amendments to the claims are made for purposes of clarity, and are not intended to alter the scope of the claims or limit any equivalents thereof. The amendments are supported by the specification and do not add new matter.

CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 925-472-5000.

Respectfully submitted,



Jason D. Lohr
Reg. No. 48,163

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 925-472-5000
Fax: 415-576-0300
Attachments
JDL:ndl
60897487 v1